ERDC MSRC Major Shared Resource Center

ERDC MSRC/PET TR/00-23

Development of Parallel 3D Locally Conservative Projection Codes for Reduction of Local Mass Errors in Hydrodynamic Velocity Field Data.

by

Mary F. Wheeler
Clint Dawson
Victor J. Parr
Jichun Li

16 June 2000

Development of Parallel 3D Locally Conservative Projection Codes for Reduction of Local Mass Errors in Hydrodynamic Velocity Field Data.

Mary F. Wheeler
Clint Dawson
Victor J. Parr
and
Jichun Li
Center for Subsurface Modeling
University of Texas, Austin

June 16, 2000

1 Introduction

This report describes the work done in a PET focused effort to develop a general parallel 3D locally conservative projection program, and as a first application, interface the software to the RMA-10 hydrodynamics simulator used at ERDC. This application of the projection program to RMA-10 hydrodynamic velocity fields allows the data to be used in water quality transport codes such as CE-QUAL-ICM with greatly reduced mass errors.

This work is based upon the UTPROJ code developed by Li, Dawson and Wheeler. The program solves for the mass correction of a 3D velocity field as the solution to an elliptic boundary value problem with appropriate boundary conditions on the hydrodynamic mesh, where the discretization is based upon the hybrid mixed-finite element method using tetrahedral, hexahedral, and prismatic elements.

This discretization of the elliptic problem is known to conserve mass element-by-element, which is why it was chosen. A discussion of the mathematical equations appears in [1].

2 UTPROJ1: Prototype Serial Version

A prototype code, UTPROJ1, was developed on serial machines. The first task was to port it to the Cray T3E, SGI Origin 2000, and IBM SP at ERDC MSRC.

To be applicable to the hydrodynamic simulator RMA-10, an interface program, the UTPROJ1 pre-processor was written in Fortran90 to convert RMA-10 data into UTPROJ1 input format.

The UTPROJ1 pre-processor defines the mixed-finite element mesh, appropriate boundary conditions for river flow boundaries, open-sea boundaries, water surface, water bottom, and shoreline boundaries, and code to build the right hand side load vector based upon the computed local mass errors of the RMA-10 flowfields for each timestep to be corrected.

Demonstration of UTPROJ1 and its pre-processor to Charlie Berger and Gary Brown of the CHL at ERDC led to a modification of the load vector to handle the volume changes in the elements in the non-steady state case where a minus volume rate change was added to the load vector.

3 UTPROJ2: Initial Parallel Version

An initial parallel version of the code, called UTPROJ2, has been developed. This effort required slightly modifying the UTPROJ1 pre-processor to partition the RMA-10 timestep data equally among a collection of processors, and a trivial addition of MPI to the code such that each processor solves the global problem for the timesteps it has been assigned. A small post-processor was written to concatenate the results into a single output file. The numerical results are identical to UTPROJ1, but UTPROJ2 has speedup equal to the number of processors used.

4 UTPROJ3: Parallel Domain-Decomposition Version

A parallel domain-decomposition version was developed from UTPROJ1 by Li and Robert McLay at the University of Texas. It is based upon the pioneering 1986 research paper [2] by Glowinski and Wheeler, which defines a non-overlapping domain decomposition method for solving elliptic problems using a mixed finite element discretization. This formulation was later modified in [3] to use a hybrid mixed finite element formulation, which allows solution of a symmetric positive definite matrix (which is preferable since the robust and efficient conjugate gradient iterative method can be used) instead of the saddle-point (indefinite symmetric) matrix required in [2].

The original paper [2] defined two dual methods, called Method1 and Method2 in the paper, where fluxes and pressures are used, respectively, on the interface faces between the subdomains. UTPROJ1 uses Method2, therefore, pressure boundary conditions are used on the interface between subdomains.

In simple terms, UTPROJ3 repeatedly solves the boundary problem of each subdomain in parallel, using zero pressure boundary values initially as a starting guess on the interfaces between subdomains, and uses a steepest descent algorithm, implemented by using conjugate gradients on the *interface operator* to compute updated pressure values on the subdomain interfaces. The UTPROJ3 algorithm, therefore, involves an outer iteration to solve the interface operator, which converges when the flux values for neighboring subdomains match to within some tolerance.

The matrix-vector multiply of the interface operator consists of solving each of the subdomain problems (with conjugate gradients), then using message-passing to swap the interface pressure values, and updating the pressure values by an averaging process. The subdomain solves are a large grain task, so the message-passing time is minimal, leading to excellent parallel speedup of the overall algorithm.

The message-passing was coded using KELP 1.3 from UCSD, which is documented in [4]. This allowed a high level programming of the message-passing part of the application. Since the message-passing in UTPROJ3 is minimal, there does not seem to be any performance problem in using the KELP interface.

5 Convergence Rate Considerations

The matrix, which arises in the formation of the mixed finite element equations to solve for the fluid velocity correction, is known to have condition number proportional to the ratio

$$(h/D)^{-2}$$

where h is the average linear dimension of an element in the finite element mesh and D is size of the entire computational domain. For example, if UTPROJ1 is used to solve a problem with 10,000 elements, the condition number of the finite element matrix would be on the order 10^8.

To speed convergence of the conjugate gradient method, one usually uses a good preconditioner to reduce the condition number of the preconditioned matrix. The current three implementations of UTPROJ use diagonal preconditioning. Without this preconditioner convergence was found to be completely erratic.

For the datasets supplied by ERDC thus far of RMA-10 data, the number of iterations to reduce the residual of the linear system to machine precision (order 10^{-16}) is between N/2 and N iterations, where N is the number of faces in the finite element mesh. We solve the sytem to such a high degree of accuracy, because we determined by numerical experimentation that the local mass errors were greatly reduced by such accurate solution, and when the linear system was solved with a final residual of 10^{-6} or 10^{-9}, the local mass errors climbed considerably.

The condition number for the interface operator (we say operator because the matrix is never formed but is only known implicitly) is proportional to the ratio

$$(h \times H)^{-1/2}$$

where h is as defined above, and H is the average linear dimension of the subdomains of the domain decomposition of the global mesh. A consequence of this formula is that each time one doubles the number of subdomains the condition number of the interface operator increases by the square root of 2.

6 Planned Enhancements to UTPROJ Implementations

Future plans include improving the convergence characteristics of the domain decomposition implementation of UTPROJ. Robust preconditioners for the subdomain and interface problems will be developed.

7 Appropriate Boundary Conditions for Velocity Correction

UTPROJ solves for the correction to the hydrodynamic velocity field. Initially it was speculated that the correct boundary condition for the correction velocity should be minus the input velocity flux, so that the corrected flux, being the sum of the input flux and the correction flux, would be zero.

More careful analysis showed that this approach is appropriate for the water bottom and land boundaries, which should have a no-flow flux condition, but is inappropriate for the water surface, which is a moving surface. We chose to use a zero flux boundary condition for the water surface velocity correction. For the river inflow boundaries, the correction velocity should have a zero flux boundary condition also, so that no modification is made to the inflow rate.

There is a problem with the entire boundary having flux boundary condition, because this defines a pure Neumann problem, which produces a singular matrix, with a one-dimensional null-space. Numerical experimentation with the pure Neumann case showed that convergence was very slow. We alleviated this problem by using zero pressure boundary conditions for the open-sea boundary for the correction velocity.

8 RMA-10 Test Cases

Gary Brown of the ERDC CHL has supplied the EQM team with five test RMA-10 datasets. One very small mesh to help debug UTPROJ, and four meshes with increasing size.

Finally, we include at the end three Tecplot graphics figures:

1. Figure 1 shows a RMA-10 test case mesh with four subdomains.

2. Figure 2 shows local mass errors before mass-error reduction.

3. Figure 3 shows local mass errors after mass-error reduction.

Figure 1 RMA-10 test mesh with four subdomains

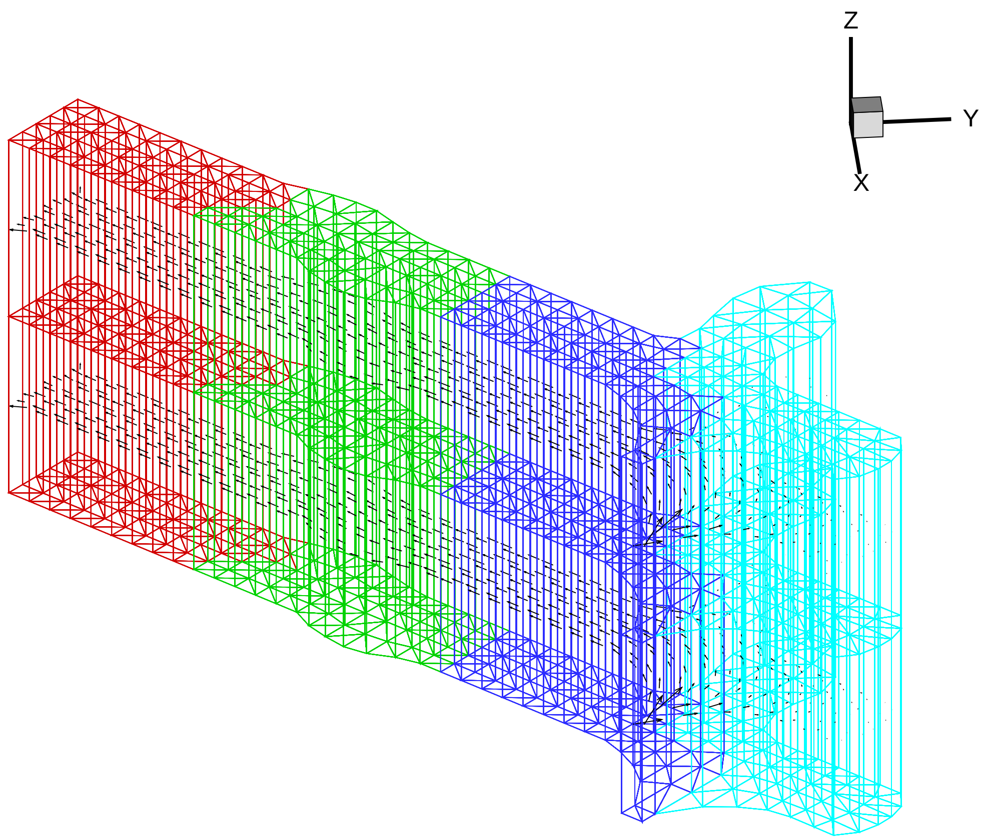


Figure 2 Local mass errors before mass-error reduction

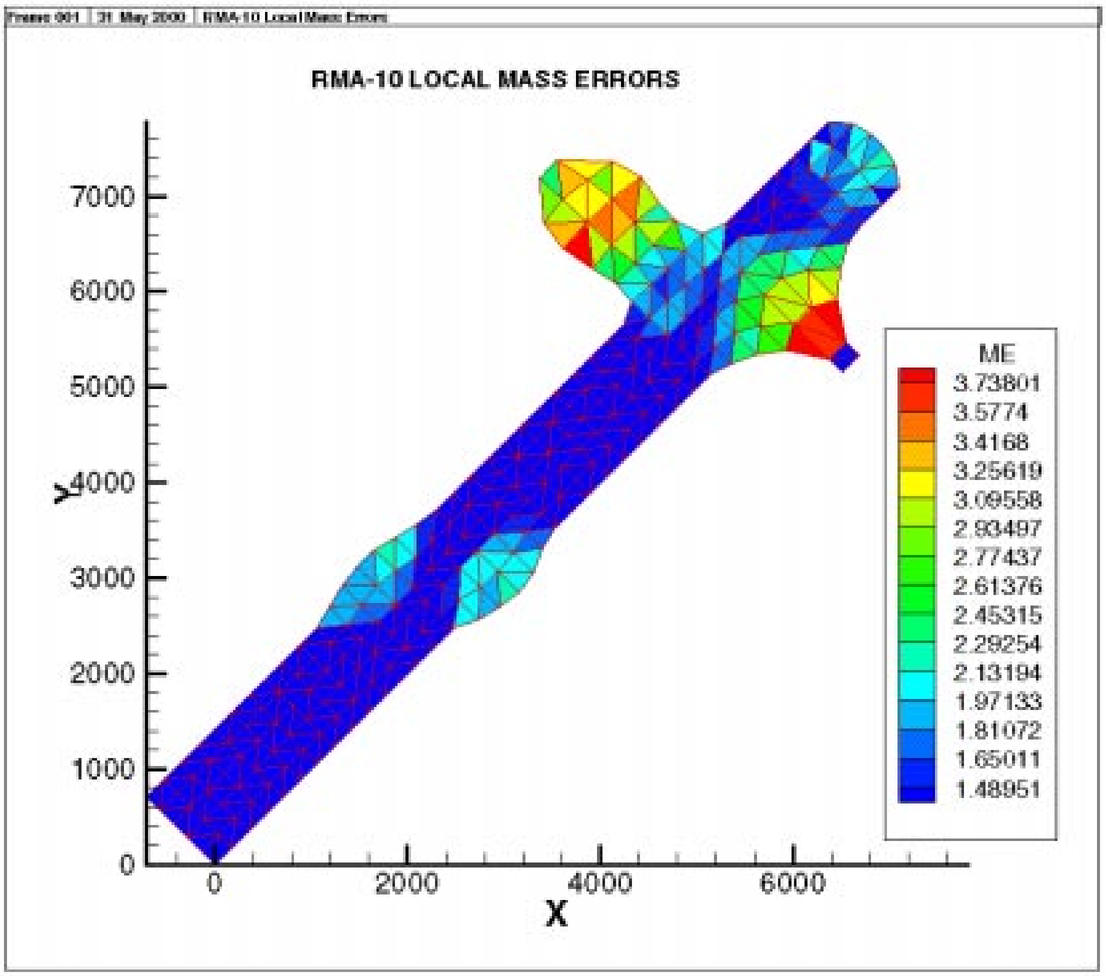
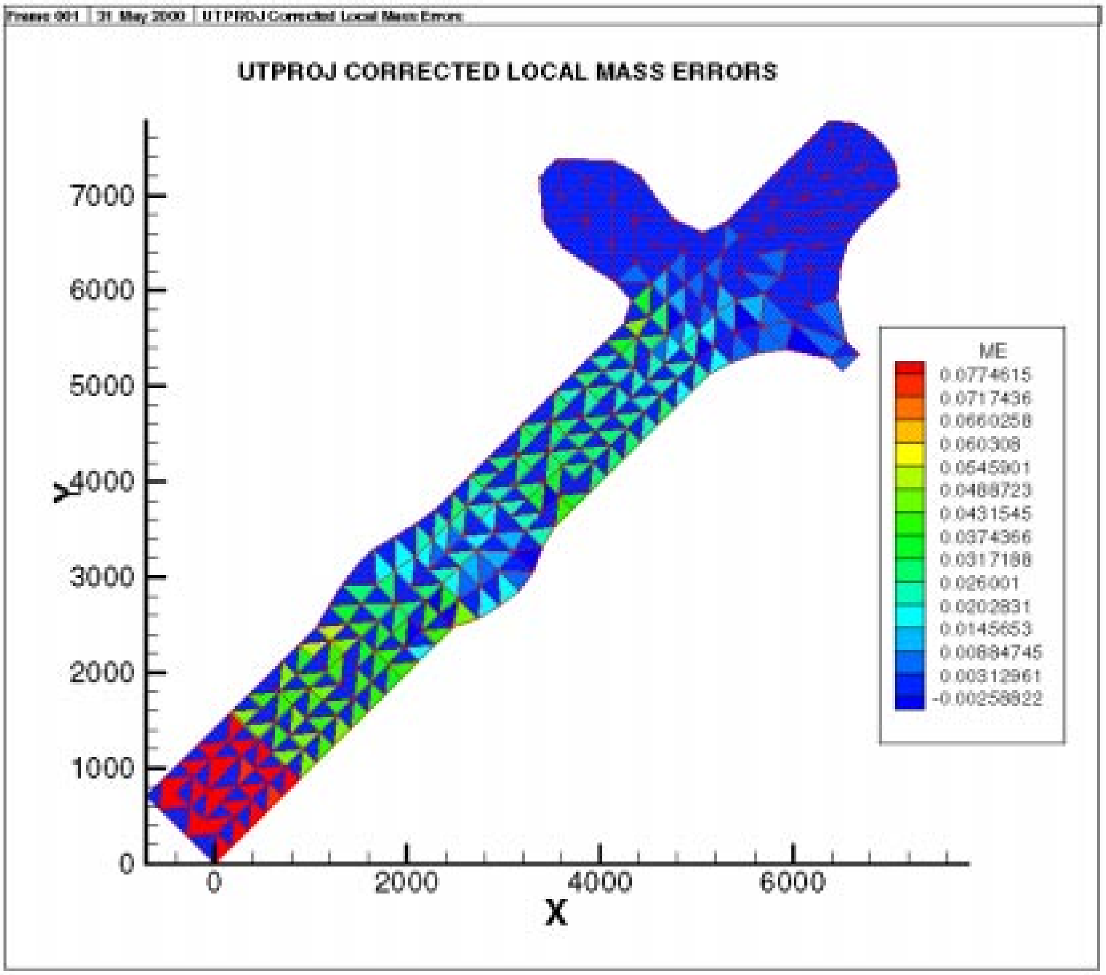


Figure 3 Local mass errors after mass-error reduction



9 Conclusions

All three implementations of UTPROJ have been shown to mass-correct the test data from RMA-10 quite adequately. However, for production work at ERDC, we believe that robust preconditioners for both the subdomain solve matrix, and the interface operator are required.

We thank Charlie Berger at ERDC for his guidance in specifying the precise requirements for RMA-10 mass error correction. We also thank Carl Cerco for his valuable input.

References

[1] S. Chippada, C. N. Dawson, M. L. Martinez and M. F. Wheeler, A Projection Method for Constructing a Mass Conservative Velocity Field, Computer Methods in Applied Mechanics and Engineering, 157, pp. 1-10, 1998.

[2] R. Glowinski and M. F. Wheeler, Domain Decomposition and Mixed Finite Element Methods for Elliptic Problems, First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski et al., eds., SIAM, Philadelphia, pp. 144-172, 1988.

[3] F. Brezzi and M. Fortin, Mixed and Hybrid Finite Element Methods, Springer-Verlag, New York, 1991.

[4] S. J. Fink and S. B. Baden, The KeLP User's Guide, v1.0, University of California at San Diego, March 1996.